bitsquare - The decentralized bitcoin exchange

Version 0.9

Motivation

Satoshi Nakamoto has created Bitcoin with the motivation to solve the problems of centralized payment systems.
It seems like an antagonism that the infrastructure of Bitcoin is dominated by centralized systems like the exchanges. It is one of the most crucial achilles heels in Bitcoins ecosystem as we could recently observe in China.
There are many discussions since a long time how to build pure P2P based solutions but unfortunately there is still no solution available.
We aim to develop such a solution which is based on pure P2P infrastructure.
We cannot prevent the involvement of banks but we can use it in a way that they are not a conscious and controlling part of the financial interaction.
A government can forbid banks to operate with Bitcoin exchanges, but they cannot censor a huge amount of individual users bank transfers.

Overview

Think of it as localBitcoin without the controlling company, so there is no central point of failure.
As there is no mutual trust between the traders we use several protection mechanisms to make sure both parties behave fair.
The P2P client software includes a Bitcoin wallet and use a secure messaging system with a distributed hash table (DHT) storage solution.
The messaging system is used for the distributed orderbook and for the trade protocol.

There will be 3 parts of the software:
- A daemon running to automate trade requests and serve as messaging network node
- A GUI client for the user interaction
- A system tray for notifications about in-coming trade requests

This **graphical overview** shows the basic high-level flow of the trade process.
Here are a few **videos** with demonstrating the current software and giving more detailed explanations.
A detailed graphical representation of the **trade protocol** should help to understand the way how the payment process will be executed.
A dedicated document discusses the possible **risks, attacks and protection mechanisms**.
An **arbitrator system** will serve as primary protection mechanism and is also described in a separate document.
Finally there is a document discussing the **funding model and organisational structure**.

Basics

- Pure P2P, cannot get shut down by a central authority
- Fiat money will be transferred from one users bank account to the others bank account without any intermediary party
- Open source, open communication, open development
- Arbitrator system as primary protection mechanism
- Collateral as additional incentive for fair behavior
- Contract holds all trade details and is signed by both traders helps in case of disputes
- Account registration with a fee and blinded bank account details attached makes sock puppet accounts more expensive and difficult
- Fraud reports as protection against bank charge backs and crime (stolen bank accounts)
- Fees as protection against fake orders and spam
- At least for the first phase there will be a limit for the trade volume, to limit the risks
- Compliant with the legal framework (at least as long Bitcoin is legal)

Limitations

- The duration for the bank transfer slows down the trade process
- Only non-reversible bank transfer types will be supported (risk of chargebacks)
- Privacy is leaked to the trading partner with the bank account details
- In case when Bitcoin is illegal it is not recommended to use it because of the risk of undercover agents
- To use the platform you need to have already Bitcoins for paying the fees (registration, offer, tx, arbitration fee) and the collateral.

Protection

For protecting against several fraud and attack scenarios we use different solutions:

- Account registration has some costs and is attached to a bank account
- Contract
- Collateral
- Arbitrator
- Fraud report
- Identity verification (optional)
- Reputation (optional, open if it will be included)

Fees

The fees are necessary for protection and it is planned that they will be used as incentive for the projects funding model. The donators will receive the fee payments and get a return for their support to build the platform. We cannot guarantee that this model will be realized but it is the

planned scenario yet. There might be reasons to change that model (legal, conceptual, technical). More details can be found in the document about funding.

Here is the range of the fees we consider.

- Account registration fee: 0.01 - 0.1 BTC
- Create offer fee: 0.1 - 1 % of trade volume
- Take offer fee: same like create offer fee
- Arbitrator base fee: Arbitrators will define that, but should be in the range of the create offer fee
- Active arbitration fee: Only in case of a dispute. Arbitrators will define that, but should be in the range of 1 - 20% of the trade volume with a min. fee in BTC (0.01 - 0.1 BTC)
- Bitcoin tx fee: Standard tx fee, e.g. 0.0001 BTC

Basic use cases

- Register the trading account
- Publish an offer
- Take an offer
- Pay the BTC funds to an escrow account
- Make bank transfer
- Release BTC funds

Extended use cases

- Register an arbitrator account
- In dispute case: Request arbitration
- In dispute case: Arbitrator release BTC funds

Technology

The proof of concept prototype is build in Java 8 with JavaFX.
As bitcoin library bitcoinJ is used. For messaging and DHT TomP2P.
It depends on further discussions and the developer team which technology we will finally use.
Python and Bitmessage might be an alternative.

Messaging system

There are 2 main use cases for the messaging system:
- Broadcast order (public data)
- Messaging between the peers (private and encrypted)

Additionally it can be used for system-wide broadcasts like:
- Fraud report update
- Alerts (updates, warnings,...)

The messaging system need to be:
- Resistant against spam/flooding
- Reliable (NAT traversal, firewalls,...)
- Protect privacy
- Possibility to store data
- Fast
- Scalable
- Support for Tor (should be possible for a later version at least)

At the moment TomP2P is used. It seems that it fulfills our requirements but it needs more investigation and review.
Here is a document with the requirements and the specification and open questions with TomP2P.

Wallet

Every trade use a different set of addresses. The anonymity should not be leaked inside the wallet (no merge of tx inputs/outputs between trades). For more anonymity external solutions (Darkwallet) should be used.

The main tasks for the wallet are:
- Manage the key pairs
- Create regular and pay to script hash (p2sh) transactions
- Sign regular and p2sh transactions
- Broadcast transactions
- Add data to a tx (OP_RETURN)

BitcoinJ is currently used as bitcoin library.

Exchange

The exchange implements the protocol for the trading process.

Protocol

The trading protocol requires that both traders software is running (background process). They don't need to be physically online but the software need to process the requests.
With broadcasting an offer the offering peer agrees to accept any take-offer request which fulfills the constraints defined in the offer.
The process until the deposit tx is published will normally take just a few seconds.
There will be a timeout for the response.
The user experience is comparable to that of a classical centralized exchange. The bank transfer is the main slowing part which can only be mitigated by using bank transfer types which

are fast. Transfers inside the same bank or the usage of modern payment processors (OKPay, Netteller, Swish, Dwolla, etc.) might speed up that bottleneck considerably.

Use case 1: Alice want to buy BTC for Fiat

Here is a detailed graphical overview of the trade protocol.

Create offer

Alice broadcasts a cryptographically signed offer to buy a set amount of BTC with a specific currency at a set rate. She also has to specify which fiat currency transfer methods and which registered arbitrators she agrees to use. The offer only reveals her registered ID, not any personal information. The offer will be stored publicly in the DHT. The offer storage is protected so that she is the only one who can change or remove her offer. There will be a max. time to live for the storage so as not exceed the DHT's storage capacity.

Orderbook

At startup every trader loads all broadcast offers for his specific currency from the DHT. The trader then filters offers for his specific fiat transfer method, and approved arbitrators, according to his preferences, and displays the result in the local order book table.

Take offer

After verification that the offer fee was paid from the offerer, Bob finds Alice's offer in the DHT, and lookups up Alice's address in the DHT to establish an encrypted direct communication channel. He needs to verify that Alice is the one who signed the offer and that the offer is still available (no other trader has taken the offer in the meantime). The offer will remain in the distributed order book until an escrow deposit is created and funded by both peers.
Then he needs to pay the take offer fee.
Until this point neither peer has revealed any private information to the other peer.

Deposit transaction

The deposit tx is created using a 2 of 3 MultiSig pay to script hash output script.
The deposit tx will be passed for completion and signing via the communication channel and will be published to the BTC blockchain by Alice.

The deposit tx contains:
Input Alice: collateral + tx fee + arbitrator base fee/2
Input Bob: collateral + tx fee + arbitrator base fee/2 + payment
Output MultiSig: 2*collateral + tx fee + payment
Output Alice (optional): change back to Alice
Output Bob (optional): change back to Bob
Output Arbitrator: arbitrator base fee
Output Data: OP_RETURN + hash of contract (20 bytes)

Contract

During the trade protocol a contract will be created, signed by both parties and its hash is attached to the deposit tx. It will serve as evidence in case of a dispute. The contract will only be stored locally by both traders, only the hash is stored in the blockchain.

Account and fee verification

During the trade protocol both traders verify the others trading account registration, the fee payments and if the other peer is not listed in the fraud report.

Fiat transaction

After the escrow deposit tx is published Alice waits for at least 1 confirmation, then she starts the bank transfer.

Create the payment tx

Alice creates the payment tx.

The payment tx looks like this:
Input: Funds from MultiSig address -> Alice signs it
Output Alice: Collateral + payment to Alice
Output Bob: Collateral to Bob

She signs her part and sends the partially signed payment tx to Bob and tells him that she has initiated the bank transfer.

Bob waits until he receives the Fiat

Bob receives the payment tx and the message from Alice that she has started the Fiat payment and will check periodically his bank account.

Bob signs and publishes the payment/refund tx

After receiving the money at the bank account, he signs the payment tx and publishes it. He can check his wallet to see that he got refunded his collateral and arbitrator fee.
For Bob all has been successfully completed.

As soon as Bob has published the payment tx Alice get a message with the tx ID and she will check her wallet until she can see that she has received the payment and her collateral.
For Alice all has been successfully completed.

Cancel offer

The creator of an offer can any time remove the offer. It will be broadcasted so all other peers will remove that offer from their orderbook.

Use case 2: Alice want to sell BTC for Fiat

The opposite situation with Alice as the BTC seller works similar. The protocol is a bit different

but there are no fundamental changes.

Use case 3: Dispute

In case that Bob does not receive the Fiat payment after the defined timeout period he can contact the arbitrator. He sends a request to the arbitrator with the contract attached and a new escrow payout tx where the arbitrators arbitration fee is included as payment to the arbitrator and where Bob gets back his payments and his collateral (and Alice's collateral).
If the arbitrator rules in Bob's favor he will receive (Alice's collateral as well as) his own collateral and payment back. So if the arbitrator decided to his favor he will be compensated for his lost trading opportunities and Alice will lose her collateral for dishonest trading.
The arbitrator has to find a fair solution and per contract terms will contact both peers and do any other contractually pre-determined due diligence.
After the arbitration the all parties can give a reputation feedback about the arbitrator. At any time if the two trading parties can agree without the arbitrator they can cancel arbitration and complete the trade themselves.

The other case that Bob has received the Fiat money but don't release the payout tx is similar. In that case Alice will contact the arbitrator after the trade timeout.
The traders could cheat to pay the arbitrator payment when publishing the payout tx, while the arbitrator has already started his arbitration work. To protect against that we could use the fraud report list.

See that document for more detailed information about the arbitrator system.

The timeout for the trade process will be defined by bank transfer type. 1 week for a national bank transfer seems to be a reasonable time but that needs more research.
There will be also some warning messages created automatically by the system to remember the user to act if needed (Alice's case: To start the bank transfer, Bobs case: To confirm that he has received the Fiat on his bank account).

TODO:
Discuss the fee and arbitration fee concept.

ID Registration

The identity registration is based on a fee payment from a registration address and use attached data (the signed and hashed bank account ids). The owner can prove ownership but cannot change or remove the onced associated data as all is stored in the block chain.

As proof of ownership and identity root we create a key pair and get a **registration address** used to pay in the registration fee (it is planned that it is paid to the donators). The fee will be a protection against sock puppet accounts and will be in the range of 0.01-0.1 BTC.

We make a transaction from the registration address with a OP_RETURN + data as output script.
Embedded data = hash(bankAccountDetails + sig(bankAccountDetails) + Messaging ID)

bankAccountDetails is a stringified list of the bank account details (IBAN + BIC + holder name + bank account type).
The private key from the EC key pair used for the registration address is used for signing.

The embedded data does not leak any private information as the bankAccountDetails cannot be derived from the hash because the sig(bankAccountDetails) is included and it is not publicly known (without signature the bank for instance could lookup if any customer has registered a trading account as the bankAccountDetails are kind of public at least to the bank).

Only when the user enters the trade protocol (after the other peer has deposited their collateral and arbitration fee to the escrow address) he will reveal his private data to the trading peer.
The peer can validate that the registration data are valid, the bank account details in the contract match the registration data and that the bank account and the users account ID are not listed in the fraud report.
See the risk analysis document for more details.

In case the user is listed in the fraud report all private data will be revealed (bank account details, account ID). Fraud report listing can only be done by arbitrators with high reputation. It contains a document for the evidence and will only be applied in clear cases of scam (bank charge back, stolen bank account, arbitrator fee fraud).

We need also to include the Messaging ID (public key) to the attached data in the registration tx. That way the Messaging ID becomes an unchangeable part of the account identity. That might be useful if we need to use the Messaging ID for protection mechanisms like exclusion from the messaging/DHT network in case of abuse (not sure how it is feasible technically and if that is needed).

Costs of account creation

A **registration fee** will be used for protection against sock puppet accounts.

The **bank account** can be considered as scarce resource.
The creation of a new bank account may produce costs, effort and takes time and there are not unlimited banks in a country available.
The banks are also exchanging information between them, so if one opens very frequently bank accounts he will make himself suspicious and can get confronted with money laundering accusations.

Offer

Alice creates an offer and need to define the trading amount. All other offer data will be derived from the preferences and the matching set with the offer.
The arbitrator will be selected in a verifiable but unbiased way, so no trader can easily force the selection to one preferred arbitrator (for potential collusion).
The selection mechanism is described in the document about arbitrators.
Also other offer constraints like supported languages, countries, bank transfer type are derived from the users settings and the offers constraint. The offer values will be an intersection of the users settings and the offers constraints.
E.g. if the user support as languages english, german and spanish and the offer only supports english, spanish and italian, then the supported languages for the trade will be the intersection between the 2 different sets which is english and spanish.

Fees

There are 3 types of fees:
- Tx fee: E.g. 0.0001 btc
- Offer creation and take offer fee: In the range of 0.001-0.01 BTC
- Arbitrator base fee: Can be chosen by the arbitrators, but will be about 0.001-0.01 BTC

To make the payment process fast we don't wait for 1 confirmation. Double spend would be possible but will be difficult and the low fee will be not much incentive to do that.
The offer creation fee will be verified by the taker, so a double spend will be visible after 1 block and would only be possible for offers which get very fast accepted.
Also there might be used a reputation system which would cover double spends of fee payments.

Collateral

The collateral will be derived from the arbitration fee which will be used only in case of arbitration. There will be a min. fee in BTC (0.01-0.1 BTC) and a percentage value from the trade volume. The collateral serves also as incentive to follow the protocol (e.g. that Bob is not lazy or careless and forget to release the payout tx), so arbitration should be only necessary in critical cases.

Contract

After offer acceptance and take offer fee payment the taker creates and signs the contract. The contract contains all relevant data about the trade, including the bank details and trading account and message ID of both traders. The contract will be verified and locally stored by both peers and will only be used and needed in case of an arbitration. The hash of the contract will be included in the deposit tx as proof that both parties have accepted the trade details.

Identity verification

If an optional identity verification might be supported is still in discussion.
See the risk analysis document for more information.

Arbitrator

See the dedicated document about the arbitration system.

Reputation system

In the risk analysis document a possible reputation system is discussed.
It will be not considered necessary for the first release.

Fraud reports

A fraud report is used against the fraud with bank chargebacks, stolen bank accounts or arbitration fee fraud.
The arbitration system will not help in those cases anymore as the BTC payment has been released already.
More details are in the risk analysis document.

Backups

Since no central server exists in this system a feature to encrypt and store a backup of all private keys, preferences, contracts, logs and histories is needed to transfer and restore this information. For instance a user should be able to backup to their Google drive account or a USB drive and restore and run their trading peer on another computer.

Limitations

In countries where Bitcoin is illegal it is not recommended to use the platform as it comes with severe risks. Undercover agents could easily find traders by acting as a peer trader.

Banks might block a bank account if they discover involvements in Bitcoin trades.
If that might be a risk in a certain environment it is recommended to use a dedicated bank account to prevent the hassles with a blocking of the primary bank account.

There will never be 100% safety when using the exchange (the same is true for centralized exchanges and any kind of money transfer).
To protect against too high losses a limit for the amx. trading volume will be used.
5-10 BTC might be considered reasonable for the first release. For the beta version probably even lower (0.5 - 1 BTC).

The privacy and anonymity protection of Bitcoin is not very high. To obtain higher level of anonymity we recommend to use solutions (Darkwallet,...) outside of the system.

Legal situation (needs confirmation from an expert)

The trade between private persons without any 3rd party included in the possession of the money should not be subject of any regulatory issues.
The arbitrators does not possess the funds at any time, so they are also not subject of any regulatory issues.

The exchange does not take fees from the trade, therefore the provider of the software is not included in the financial interaction and not subject of regulations nor can be considered as business.
There is also no automated order matching and therefore no regulation issues.

The arbitrators will take a fee and can be therefore considered as professional businesses. It is in their responsibility to act according to the legal and fiscal requirements.
The platform does not earn money for supplying the infrastructure and has no business relationship to them.

Funding Model

Please note:
The funding model referenced in that paper is not considered anymore to be used due to legal uncertainties.

Development

The development process will be open, transparent and will be organised in a democratic structure. All code is open source (AGPL).

Possible future extensions

The basic structure of the exchange can also be used for a crypto-crypto exchange and a P2P marketplace.

A crypto-crypto exchange could be build in a real trustless way, so no arbitration and none of the other protection mechanisms are needed. The main problem would be how to connect without too much effort to a huge variety of different alt coins and systems.

An interesting aspect of an crypto-crypto exchange would be the anonymisation of digital currencies.

P2P marketplace:
Change the Bank transfer with a mail delivery and the exchange becomes to a marketplace.
A P2P marketplace would introduce the need for a robust reputation system because trading products cannot be reduced to a binary result like in a fiat exchange: money received or not. The success of a product trade can have a variety of satisfaction levels: Product quality, expectation, description, delivery,...
So a reliable reputation system is essential but has its open problems.
Also the huge variety of products introduce considerable effort for the presentation layer.